AMENDMENTS TO THE CLAIMS:

1. (Currently Amended) A method of maintaining the memory contents of a memory medium having a plurality of memory sectors that are erased before being rewritten, the method comprising the steps of:

maintaining a plurality of data structures in memory, each data structure corresponding to a flash sector, and each data structure recording characteristics of a data access to the corresponding flash sector as well as containing a cached copy of file-system blocks present in the corresponding flash sector. ~~independent sector caches each respectively corresponding with a stored memory sector of the memory medium;~~

~~writing data to one or more of said sector caches with required changes to the corresponding stored memory sectors of the memory medium; and~~

~~accessing the contents of said stored memory sectors from the corresponding sector caches, if said contents are stored in said corresponding sector caches.~~

2-4. (Cancelled)

5. (Currently Amended) The method as claimed in claim 1 [[4]], further including the step of: determining a sector weightage for each cached sector that is dependent upon a combination of number of sector switches suffered by the corresponding flash sector and a number of dirty file-system blocks present in the corresponding flash sector, and wherein said sector weightage is dynamically recomputed; ~~maintaining a record, for each of the sector caches, of the number of dirty blocks for which the contents of a sector cache and the corresponding stored memory sector are not currently equivalent~~.

6. (Currently Amended) The method as claimed in claim 5, further comprising the step of: incrementing the sector weightage whenever there is a write access to a block within the cached sector, as long as said block has not already caused the sector weightage to be incremented

~~recorded number of dirty blocks for a sector cache after performing said step of writing data to the corresponding sector cache in respect of said stored memory sector, if the block for which said step of writing data is performed is not currently marked as dirty~~.

7. (Original) The method as claimed in claim 6, further comprising the step of: maintaining a record, for each of the sector caches, of the number of sector switches that represent instances in which said step of writing data to the cached memory sectors successively occurs for different sector caches.

8. (Original) The method as claimed in claim 7, further comprising the step of: incrementing the recorded number of sector switches in respect of a sector cache after performing said step of writing data to the sector caches, if said step involved a sector switch.

9. (Original) The method as claimed in claim 8, further comprising the step of: recording a predetermined threshold number of sector switches.

10. (Original) The method as claimed in claim 9, further comprising the step of: resetting the recorded number of sector switches to zero once the recorded number of sector switches exceeds the predetermined threshold number of sector switches.

11. (Currently Amended) The method as claimed in claim 10, further comprising the step of: resetting the sector weightage of the accessed sector to zero and flagging all file-system blocks in the said sector as non-dirty irrespective of whether the contents of the blocks in the sector cache are different from the corresponding flash memory locations, once the recorded number of sector switches exceeds a predetermined threshold number of sector switches ~~recorded number of dirty blocks to zero once the recorded number of sector switches exceeds the predetermined threshold number of sector switches~~.

12. (Original) The method as claimed in claim 11, further comprising the step of: receiving a request to perform a write operation in respect of a stored memory sector.

13. (Original) The method as claimed in claim 12, further comprising the step of: determining whether the stored memory sector subject of said received write request is stored in a sector cache.

14. (Original) The method as claimed in claim 13, further comprising the step of: identifying the sector cache corresponding with the stored memory sector subject of said received write request, if the stored memory sector subject of said received write request is stored in the sector cache.

15. (Original) The method as claimed in claim 14, further comprising the step of: writing to the identified sector cache, in response to said received write request, if the stored memory sector subject of said received write request is stored in the sector cache.

16. (Original) The method as claimed in claim 13, further comprising the step of: determining whether there is an available sector cache not currently associated with a corresponding stored memory sector, if the stored memory sector subject of said received write request is not stored in a corresponding sector cache.

17. (Original) The method as claimed in claim 16, wherein if there is not an available sector cache, the method further comprising the steps of:

erasing a selected one of the stored memory sectors of the memory medium; and

writing data from the corresponding sector cache to said selected stored memory sector of the memory medium with the contents of the corresponding sector cache;

wherein the corresponding sector cache becomes available to accept write requests in respect of different stored memory sectors.

18. (Currently Amended) The method as claimed in claim 17, wherein said selected one of the stored memory sectors is selected on the basis of the corresponding sector cache that has the highest sector weightage ~~greatest number of recorded dirty blocks~~ of all the sector caches.

19. (Original) The method as claimed in claim 18, further comprising the steps of:

reading the contents of the stored memory sector into the corresponding sector cache; and

writing data to the corresponding sector cache.

20. (Currently Amended) The method as claimed in claim 11, wherein sectors holding critical meta data information are automatically detected and offered persistent storage ~~1, wherein one or more reserved sector caches sectors persistently correspond with respective stored memory sectors of the memory medium~~.

21 - 23. (Cancelled)

24. (Currently Amended) A device driver for a memory medium device having a plurality of memory sectors that are erased before being rewritten, the device driver comprising:

code means for maintaining a plurality of data structures in memory, each data structure corresponding to a flash sector, and each data structure recording a copy of file-system blocks present in the corresponding flash sector as well as a sector weightage that is dependent upon a combination of a number of sector switches suffered by the corresponding flash sector and a

number of dirty file-system blocks present in the corresponding flash sector, and wherein said sector weightage is dynamically recomputed;

code means for efficiently managing the sectors in the cache using the

said sector weightage that is dynamically recomputed ~~independent sector caches each respectively corresponding with a stored memory sector of the memory medium~~

~~code means for writing data to one or more of said sector caches with required changes to the corresponding stored memory sectors of the memory medium; and~~

~~code means for accessing the contents of said stored memory sectors from the corresponding sector caches, if said contents are stored in said corresponding sector caches~~.

25. (Currently Amended) A computer system operating under direction of an operating system and having an associated memory medium device having a plurality of memory sectors that are erased before being rewritten, the operating system interacting with the memory medium device via a device driver, the device driver comprising:

code means for maintaining a plurality of data structures in memory, each data structure corresponding to a flash sector, and each data structure recording a copy of file-system blocks present in the corresponding flash sector as well as a sector weightage that is dependent upon a combination of a number of sector switches suffered by the corresponding flash sector and a number of dirty file-system blocks present in the corresponding flash sector, and wherein said sector weightage is dynamically recomputed;

code means for efficiently managing the sectors in the cache using the said sector weightage that is dynamically recomputed ~~independent sector caches each respectively corresponding with a stored memory sector of the memory medium;~~

~~code means for writing data to one or more of said sector caches with required changes to the corresponding stored memory sectors of the memory medium; and~~

~~code means for accessing the contents of said stored memory sectors from the corresponding sector caches, if said contents are stored in said corresponding sector caches.~~

26-27. (Cancelled)